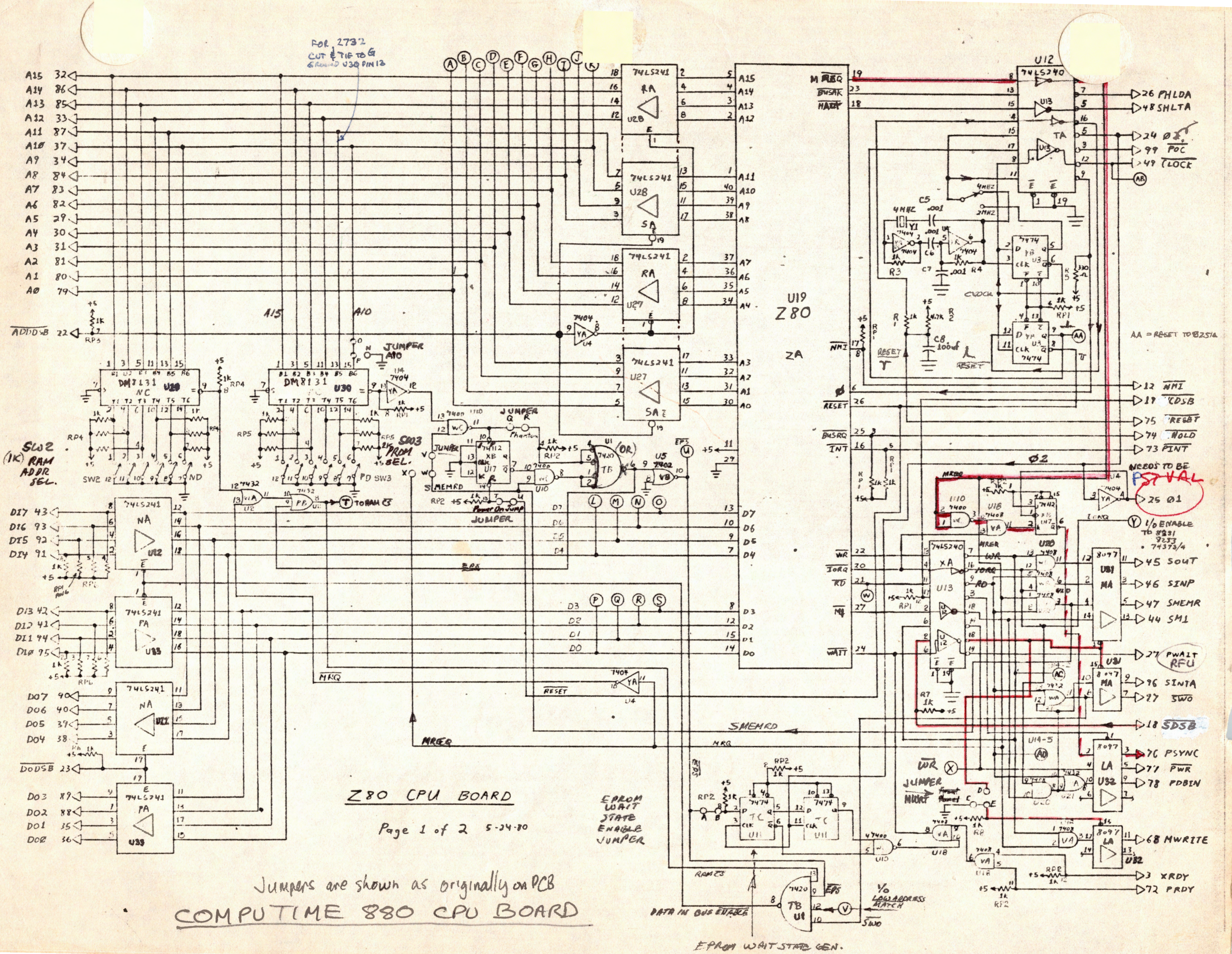
SBC880

SINGLE BOARD COMPUTER
REFERENCE MANUAL

Also = QT Computer Systems SBC+2/4
15620 South Inglewood Ave
Lawndale 90260 213) 970 0952
800) 421 5150

Dealers PriorityOne LA
Computime LA
~~Santa Rosa Comp. Ctr.~~
ASAP LA

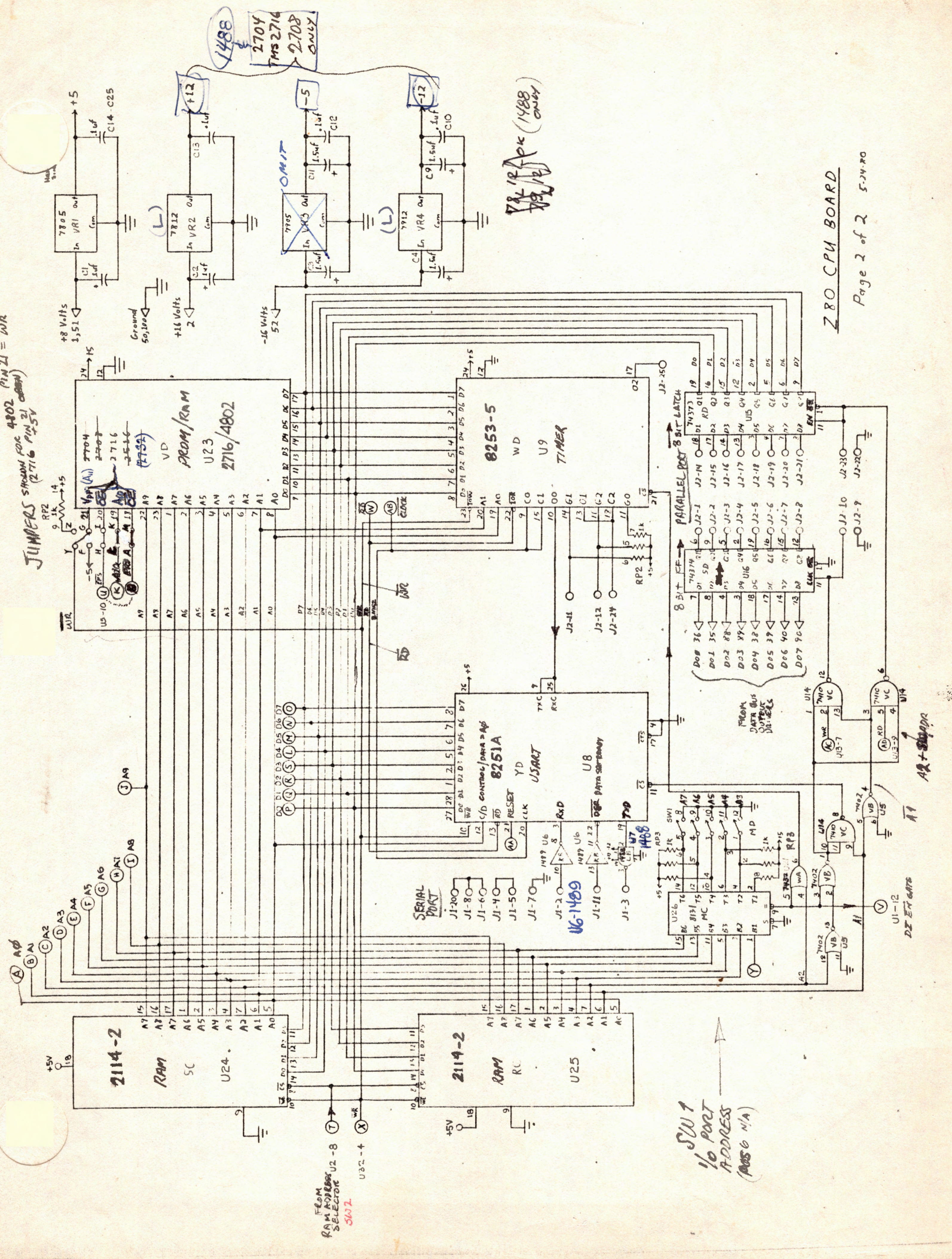


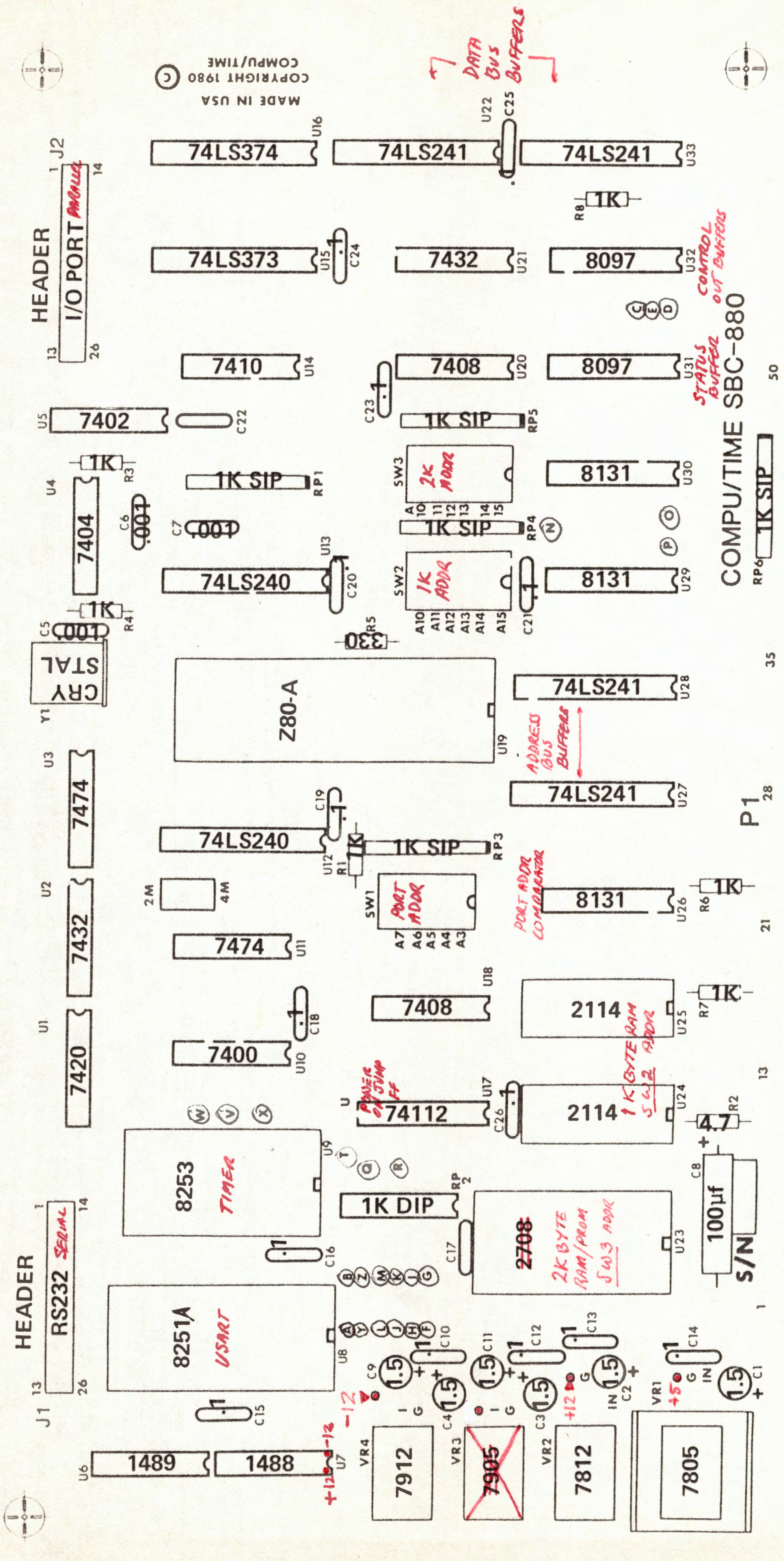
Z80 CPU BOARD

Page 1 of 2 5-24-80

Jumpers are shown as originally on PCB

COMPUTIME 880 CPU BOARD


Z80 CPU BOARD
Page 2 of 2
5-24-80
7805
VR1
7812
VR2
7905
VR3
OMIT
7912
VR4
(L)
+5
+12
-5
-12
1488
2704
TMS 2716
2708
ONLY
1488 ONLY
+8 Volts 1,51
+16 Volts 2
Ground 50,100
-16 Volts 52
C14-C25
PROM/RAM
U23
2716/4802
2704
2708
2716
2532
2732
JUMPERS SHOWN FOR 4802 (2716 PIN 21 = +5V)
PIN 21 = WR
8253-5
WD
U9
TIMER
74373
8 BIT LATCH
U15
PARALLEL PORT
74374
8 BIT FF
U16
FROM DATA BUS OUTPUT DRIVERS
8251A
YD
USART
U8
DSR DATA SET READY
1489 U6
1488 U7
U6-1489
SERIAL PORT
SW1
RP3
U26
8131
MC
7410
U14
7402
U5
A2+BOARD
DI EN GATE
U1-12
2114-2
RAM
SC
U24
RC
U25
FROM RAM ADDRESS SELECTOR SW2
U2-8
U32-4
SW1
I/O PORT ADDRESS
(PORT 6 N/A)
CLOCK
WR
RD



SILKSCREEN ART

SIZE	CODE IDENT NO	DRAWING NO
C		SBC-880

SCALE 2X SHEET 4 of 4

TABLE OF CONTENTS

Page Number

SECTION I

INTRODUCTION

The SBC 880 Processor Board is a powerful Z80 based design which is compatible with the proposed IEEE S-100 bus standard. The SBC 880 contains enough features to allow its use as a stand alone single board system or as the main CPU board in a larger system.

The SBC 880 is manufactured using the finest quality components that are conservatively rated to assure long life. Assembled and tested boards are burned in and rigorous tests with a real time disk system are performed to assure that the board will work properly when you receive it.

If you have purchased a kit, we suggest that you read the manual in its entirety before attempting to assemble your board. If you follow the assembly instructions step-by-step, the construction of your board will proceed more smoothly and you will be less prone to assembly errors.

SECTION II

FEATURES

An on board EProm can be addressed on any 1K or 2K boundary. A 1K by 8 static ram may be used in the place of the EProm if desired. Power on jump is available directly to the EProm (or ram). An optional wait state can be enabled for the on board EProm (or ram). The on board EProm may optionally be used in shadow mode to allow the use of 64K or more of Ram. Devices that can be used in the EProm location are the 2708, 2716 EProms or the 4118 static ram. In addition to the EProm location, an additional 1K of static ram is provided that can be located on any 1K boundary. If the EProm is not used and the static ram is used instead at the EProm location, a total of 2K bytes of ram may be present on the CPU board.

The SBC 880 is equipped with a USART and a RS232 interface. The baud rate is programmable by means of a programmable timer. All model signals required by terminal type of equipment are provided for by the CPU board and terminal equipment may be connected directly to the RS232 connector. Reverse channel capability is available for use with buffered types of devices, such as printers. The reverse channel is occasionally needed as a busy or ready indication from the connected device, such as, out of paper or ribbon. An on board crystal provides all system timing and switch selectable 2 or 4 MHz operation .

DMA capability is provided as well as a means of having the MWRT signal generated on the CPU board or elsewhere in the system under control of DMA logic or a front panel.

Two programmable timers are available for use by programs run with the SBC880. The timer output and controls are available at the parallel 10 connector. A parallel input and a parallel output port are available for use on the CPU board.

All S-100 bus signals are fully buffered and regulators are used for all on board voltages to assure an electrically clean and stable design. A top quality PC board is used with solder mask on both sides, gold plated contacts and plated through holes.

SECTION III

FUNCTIONAL DESCRIPTION

Refer to the SBC 880 schematic while reading the descriptions of each functional block that follows.

Z80A CPU

The SBC 880 uses a powerful Z80 CPU that is capable of being run at 4 MHz.

OSCILLATOR

A crystal controller circuit provides all the timing for the S-100 bus, the Z80 CPU, and the Baud Rate Timer, and the two programmable timers. Associated with this circuit are a wait state generator for the on board EProm and reset circuitry which also generates a power on clear signal.

STATUS AND CONTROL BUFFERS

The status and control buffers interface the Z80 CPU status and control signals to the S-100 bus. These buffers may be tri-stated by a DMA device to allow a transfer of data between the DMA device and memory. The DMA device assumes control on the status and control lines for the duration of the DMA transfer. When a DMA access is requested by activating the Z80 DMA request signal, the acknowledgment signal from the CPU is made available on the S-100 bus.

ADDRESS BUFFER

The address buffer is a 16 bit tri-state buffer which drives the Z80 CPU's sixteen address bits to the S-100 bus and to other circuitry on the SBC 880 CPU board. It is also tri-stated by DMA devices when a transfer of data is to occur. the DMA device will then provide the address for the duration of the data transfer.

DATA OUT BUFFER

The data out buffer is an 8 bit tri-state buffer which drives the eight data bus signals from the Z80 CPU to the S-100 data out bus. The data out bus only contains valid data during memory write or I/O output cycles. The data out bus will be tri-stated by DMA devices when they are transferring data to memory.

DATA IN BUFFER

The S-100 data in bus is provided to the Z80 CPU during memory read or I/O input cycles to devices external to the SBC 880 CPU board. The data in buffers are disabled during memory write or I/O output cycles when the Z80 CPU is driving data to the data out buffers. The data in buffers are disabled whenever devices on the SBC 880 CPU board are being accessed to allow the device being accessed to place data on the Z80 CPU data bus.

MEMORY DECODE AND CONTROL

The memory decode and control circuitry decodes the high order address bits and selects the EProm or the static ram which is located on the CPU board.

EPROM

The on EProm can be a 1K (2708 type), 2K (2716 type), or a 1K by 8 static ram (4118 type). The EProm may be selected on any 1K or 2K boundary and the optional ram can be selected on any 1K boundary.

1K by 8 STATIC RAM

2148 @ 70nS

The on board ram may be selected to any 1K boundary. The type of ram provided is completely static (2114 type) and requires no refresh. This ram may be used to hold the stack when running diagnostic tests on a bad dynamic ram board or it allows the use of the SBC 880 as a stand alone system.

I/O DECODE AND CONTROL

The I/O decode and control circuitry decodes the lower 8 bits of the address bus to determine when the USART, timer or parallel I/O ports on the SBC 880 are being addressed for input or output operations.

SERIAL I/O

The serial I/O provides asynchronous communication via an RS232 interface. The baud rate is provided by a programmable timer. The USART is an 8251 type.

PROGRAMMABLE TIMERS

Two programmable timers are available for use on the SBC 880. An 8253 timer chip is used and the timers are clocked from the crystal controlled clock oscillator on the SBC 880 board.

PARALLEL I/O PORTS

An 8 bit parallel output port and an 8 bit parallel input port are provided on the SBC 880 board. The ports are implemented with TTL type circuitry (74LS373 and 74LS374).

SECTION IV

ADDRESS BUS

The address bus of the Z80 CPU is buffered using 74LS241 devices. During DMA operations the DMA device will drive S-100 pin 22 (ADDSB/) low to tri-state the address bus drivers. The DMA device can then place its own address on the S-100 bus.

DATA IN BUS

During memory read or I/O input operations, the S-100 data in bus is received and driven to the Z80 bi-directional data bus by a 74LS241 type device. Circuitry is provided (1/2 of 7420) to disable the data in buffers under the following conditions:

1) On board EProm or optional ram selected
2) On board static ram selected
3) Programmable timer selected
4) Parallel I/O port selected
5) Memory write or I/O output operation in progress

DATA OUT BUS

The Z80 CPU bi-directional data bus is driven to the S-100 data out bus by 74LS241 type buffers. The buffers provide write data from the Z80 to devices on the S-100 bus during memory write or I/O output operations. A DMA device wishing to transfer data on the data out bus will drive pin 23 (DODSB/) on the S-100 bus to a low state. This will disable the data out buffers on the SBC 880 board and allow the DMA device to place data on the bus.

STATUS SIGNALS

Status signals SM1, SMEMR, SINP, SOUT, SINTA, and SWO/ are provided to the S-100 bus by a 8097 type tri-state buffer. A DMA device may drive pin 18 (STATDSB/)

low to tri-state the status buffer to allow the DMA device to gain control of the bus. Control signals PSYNC, PWR/, and PDBIN are provided to the S-100 bus by one section of a 8097 type tri-state buffer. A DMA device may drive pin 19 (C/CDSB/) low to disable this buffer and gain control of the bus. The Mwrite signal is provided to the bus by the other half of the 8097 buffer that was used by the control signals. This buffer may be controlled in several ways. The SBC 880 board as you received it has this buffer permanently enabled so that the CPU board is always the source of the Mwrite signal. If your system contains another device that is to be a source for the Mwrite signal, then you may cut the etch between points C and E to disable the buffer on the SBC 880 board. If desired, the Mwrite buffer may be disabled by the STATDSB/ signal when a DMA device is using the bus. This option is enabled by cutting the etch between points C and E, and then installing a jumper between points D and E. This option requires that the DMA device have a buffer to provide the Mwrite signal that is enabled by the STATDSB/ signal or a signal with the same timing. This is necessary to prevent floating the Mwrite signal on the S-100 bus while transferring control of the bus. Data in memory may be overwritten if this signal is left floating.

MWRITE should be pulled low @ bus terminator?

The signal SOUT is active (high) when WR/ and IORQ/ signals are active (low) at the Z80 CPU. SINP is active (high) when the signals RD/ and IORQ/ are active (low) at the Z80 CPU. SMEMR is active (high) when the signals RD/ and MREQ/ are active (low) at the Z80 CPU. MWRITE is active (high) when signals WR/ and MREG/ are active (low) at the Z80 CPU. SM1 goes active (high) when M1/ goes active (low) at the Z80 CPU.

OTHER STATUS SIGNALS

SINTA goes active (high) when M1/ and IORQ/ go active (low) at the Z80 CPU. SWO/ goes active (low) when the Z80 CPU is not performing any input cycles. This signal is used to provide an early indication that a write or output cycle is going to take place. SWO/ is active (low) when RD/ is inactive (high) or SINTA is inactive (low). SINTA will be inactive (high) at the Z80 CPU. The signal SXTRQ/ is not generated by the SBC880 board because only 8 bit data is required by the Z80 CPU. SXTRQ/ is used to request 16 bit data transfers. SHLTA goes active (high) when HLTA/ goes active (low) at the Z80 CPU

CONTROL SIGNALS OUTPUT

The signals PSYNC, PWR/ and PDBIN are tri-stated when C/CDSB/ goes active (low). C/CDSB/ will be driven low by a DMA type device when the device wants to take control of the bus.

The PSYNC signal goes active (high) momentarily at the start of any valid I/O or memory cycle. The timing for this signal is developed by a flip flop and produces timing as defined by the IEEE S-100 specification. The PSYNC signal is not produced during Z80 memory refresh cycles. PWR/ goes active (low) when WR/ goes active (low) at the Z80 CPU. The signal PDB1N goes active when a memory read of I/O input cycle is in process or when an interrupt is being acknowledged (SINTA high). PDBIN is active (high) when RD/ is active (low) at the Z80 CPU or when SINTA is active (high). PHLDA goes active (high) when BUSAK/ goes active (low) at the Z80 CPU. This signal acknowledges a DMA request and indicates that the requesting device with the highest priority may take control of the bus. The Z80 generates this signal in response to the signal BUSRQ/ going active (low). BUSRQ/ goes active (low) when a DMA device drives the S-100 signal PHOLD/ active (low). BUSAK/ only goes active when BUSRQ/ has gone active and the Z80 CPU is at a point in its operation where a DMA access can be performed properly. PHLDA/ is always driven and cannot be tri-stated.

CONTROL SIGNALS - INPUT

The IEEE S-100 bus signal SIXTN/ is not used by the SBC880 because only 8 bit accesses are required by the Z80 CPU. The signal SIXTN/ is a response to a request for a 16 bit memory access. Since a 16 bit access is never requested by the SBC880, this signal is ignored. The signals XRDY or PRDY when driven low will make the WAIT/ signal go active (low) at the Z80 CPU. The EProm wait state generator can make the WAIT/ signal active for one clock cycle during accesses to the on board EProm. The EProm wait state generator must be enabled by a jumper option. PRDY/ is normally used by slow memory or I/O devices to extend an access cycle by inserting wait states at the Z80 CPU. The device being accessed holds PRDY/ active (low) for the number of clock cycles (wait states) desired. XRDY is normally used by front panel type devices to halt or single step the processor. PWAIT/ goes active (high), WAIT/ goes active (low). When PINT/ is driven active (low) at the S-100 bus the signal INT/ will be active (low) at the Z80 CPU. This is the maskable interrupt request input to the Z80. When NMI/ is driven active (low) at the S-100 bus the signal NMI/ goes active (low) at the Z80 CPU. This is the non-maskable interrupt request input to the Z80.

When PHOLD/ is driven active (low) at the S-100 bus the signal BUSRQ/ will be active (low) at the Z80 CPU. The PHOLD/ signal is used by DMA devices to request access to the bus.

DMA CONTROL LINES

The primary lines used to tri-state the SBC880 bus drivers for DMA operations

DODSB/, ADDSB/, STATDSB/ and C/CDSB/. DODSB/ tri-states the data out bus drivers when it is driven active (low). ADDSB/ tri-states the address bus drivers when it is driven active (low). STATDSB/ tri-states the status signals SOUT, SINP, SMEMR, SM1, SINTA and SWO/ when it is driven active (low). MWrite may be tri-stated by STATDSB/ if selected by a jumper option. C/CDSB/ tri-states the signals PSYNC, PWR/ and PDBIN when driven active (low). When a DMA device is granted access to the bus by the PHLDA signal going active (high) it will normally activate the DMA control signals and drive its own signals on to the bus.

SYSTEM POWER LINES

A positive 8 volts DC should be present on S-100 pins 1 and 51. This voltage is regulated on the SBC 880 to develop +5 volts. The regulator is decoupled on its input by a 1.5 uf capacitor and the +5 volt ouput is decoupled by .1 uf capacitors at various places around the board.

A negative 16 volts DC should be present at S-100 pin 52. This voltage is regulated by two regulators on the SBC 880 to develop -12 volts and -5 volts. Both regulators are decoupled with 1.5 uf capacitors at their inputs and outputs. A positive 16 volts DC should be present at S-100 pin 2. This voltage is regulated on the SBC 880 to develop +12 volts. The regulator is decoupled by a 1.5 uf capacitor at its input and a .1 uf capacitor at its output. Ground for all of the DC suppliers should be present on S-100 pins 50 and 53.

±12 1488 only
-5 not used

SYSTEM CLOCK

The SBC 880 generates all timing from a 4 MHz crystal controlled oscillator. The 4 MHz clock is divided down to 2 MHz by a flip flop. The 2/4 MHz switch selects which clock rate is applied to the Z80 CPU and the related circuitry. The selected clock is provided to the S100 bus as Ø2. An inverted version of Ø2 is provided as Ø1. The 2 MHz clock is provided directly to S100 bus as CLOCK/. This clock is always 2 MHz and is not affected by the 2/4 MHz switch.

PIN 25; SHOULD BE PSTVAL

SYSTEM RESET FUNCTIONS

When PRESET is driven active (low) at the S100 bus a 100 uf capacitor on this line is discharged. This signal is normally driven low by the system reset button being pressed. The PRESET signal is synchronized to the system clock with a flip flop and is then applied to the following circuits:

1) The Z80 CPU
2) The 8251 USART
3) POC at the S100 bus
4) The power on jump latch

The reset signal will remain active (low) after switch is released for approximately 470 milliseconds due to the time it takes to charge the 100 uf capacitor to a true level. This same circuit de-bounces the reset switch and provides a reset signal during power up.

SECTION V   INTERFACES

EPROM INTERFACE

The EProm will be selected for access under the following conditions:

1) When power on jump is enabled (I to U present) and the power on jump latch is set. The EProm is unconditionally selected until the latch is reset. The power on latch is set any time the system is powered up or the system reset button is pressed. The power on latch is reset when a memory read operation is performed and the address being read ~~compares to~~ matches the switch settings of the EProm select switches. For 1K x 8 EProm (2708 type) the 8131 comparator compares address bits 10 thru 15 to the EProm switches. For 2K x8 EProm (2716 type) the 8131 comparator compares address bit 11 thru 15 to the EProm switches.

2) When not using the phantom option (Q to R present) and a memory read operation is performed with an address that ~~compares with~~ matches the EProm switch settings (as detected by the 8131 comparator).

The EProm may optionally be replaced by a 4118 RAM. When the RAM is used the MEMRD signal is replaced by MRQ so that the RAM signal is accessed during memory read and memory write operations. The Z80 WR/ signal is jumpered to the RAM WE/ input to allow the Z80 to write data into the RAM during memory write cycles when the RAM is selected.

The EProm or optional RAM is directly connected to the Z80 bi-directional data bus and only appears on the S100 bus indirectly through the Data Out bus drivers. Note that only Z80 CPU can directly access the EProm or optional RAM. It is not necessary to have the S100 bus in operational condition to successfully access the EProm or optional RAM. ~~It is not necessary to have the S100 bus in operational condition to successfully access the EProm or optional RAM~~. This feature allows diagnostic programs to be run in EProm to diagnose a failing S100 bus. The SBC 880 can communicate~~d~~ with a console device and run diagnostic tests even when the S100 bus is completely inoperative.

PHANTOM MODE

When the phantom EProm option is used (Q to R cut) the EProm is only selected after a power up or when the system reset is pressed and the power on latch is set. The EProm will be selected for all memory read operations that occur while the power on latch is still set. During this time, memory write operation will address memory external to the SBC 880 in a normal fashion. Likewise, I/O input and output cycles are unaffected by the power on latch. Therefore, the program in the EProm can be used to boot data from an I/Odevice into memory after a power up or system reset operation. The EProm select switches can then be set to detect the starting address of the code that the EProm program boots into memory. A jump to the starting address will then be detected by the 8131 comparator and will reset the power on latch. When the power on latch is reset, the EProm can no longer be accessed (the comparator cannot select the EProm because Q to R is open). The data in memory is now accessed in a normal fashion and the EProm effectively disappears from the system until needed at the next power up or system reset operation. (ONLY)

RAM INTERFACE

A 1K x 8 block of static RAM is implemented using two 2114 RAM chips on the SBC 880. The RAM is selected by a 8131 comparing address bits 10 thru 15 to the RAM select switches. When a memory operation is performed by the Z80 CPU (MRQ active) and the comparator detects a match between the address and the RAM switches, the CS/ lead goes active (low) at the RAM chips. When a memory write operation is performed the Z80 WR/ signal is active (low) at the WE/ inputs of the RAM chips to allow the Z80 to write data into the RAM. The RAM on the SBC 880 is directly connected to the Z80 CPU bi-directional data bus and can only be accessed directly by the Z80 CPU. The RAM data is indirectly available at the S100 data out bus. The SBC 880 RAM, EProm and USART will function independently of the S100 bus and this allows diagnostics to be performed by the SBC 880 when the S100 bus is inoperative. An example of this would be running a memory diagnositc on the SBC880 while diagnosing a dynamic RAM board that contains the balance of the system's memory. The diagnostic programs can use the on board static RAM for scratchpad and stack operations. The diagnostic routines would run properly even if the RAM board being diagnosed was affecting signals on the S100 bus. In normal use, the on board RAM may be located in the same address space as other memory in your system with no conflicts between the memory devices. This will be necessary if your system uses a full 64K of RAM. Whenever the on board RAM is accessed during a memory read operation the S100 bus data in bus receivers are disabled and the on board static RAM is allowed to supply data directly the Z80 CPU. Thus an external device responding to the same memory cycle would have its data ignored by the SBC 880 and the on board RAM would supply data instead. When your system RAM is inoperative, the SBC 880 on board RAM will be available to get your system going again.

I/O INTERFACE CIRCUITRY

The Z80 I/O devices are selected by an 8131 comparator. The 8131 compares address bits A3 thru A7 to the I/O select switches and looks for IORQ to be active (high) indicating than I/O access is in process. The Z80 uses address bits A0 thru A7 to address I/O devices. Address bits A3 thru A7 are tested by the 8131 comparator and address bits A1 and A2 are decoded with gates to select the individual I/O devices on the SBC 880 as follows:

		4	2	1			
RD	WR	A2	A1	A0	Device Selected	Operation	
1	0	0	0	0	8253 Timer	Read baud rate time	
0	1	0	0	0	" "	Write baud rate time	
1	0	0	0	1	" "	Read counter 1	
0	1	0	0	1	" "	Write counter 1	
1	0	0	1	0	" "	Read counter 2	
0	1	0	1	0	" "	Write counter 2	
1	0	0	1	1	" "	Illegal	
0	1	0	1	1	" "	Write mode word	
1	0	1	0	0	Input Port	Read input port	
0	1	1	0	0	Output Port	Write output port	PARALLEL
1	0	1	0	1	Input Port	Read input port	
0	1	1	0	1	Output Port	Write output port	
1	0	1	1	0	8251 USART	Read data register	
0	1	1	1	0	" "	Writer data register	SERIAL
1	0	1	1	1	" "	Read status register	
0	1	1	1	1	" "	Write control register	

The final I/O port address for each device is determined by the I/O select switches which determine what state of address bits A3 thru A7 will cause the I/O devices on the SBC 2/4 to be selected. Note that for some devices during an input cycle (RD active) a different operation takes place than for an output cycle (WR active) at the same address. When A2 is high and A1 is low, an input cycle selects the input port and an output cycle selects the output port. Address bit A0 is ignored when accessing the parallel input and output ports and the same devices are selected with A0 high or low.

8253 TIMER INTERFACE

The SBC 880 provides a 2 Mhz clock from the clock oscillator circuitry to the count inputs of timer zero and timer one of the 8253. Timer ØØ zero is used as a programmable baud rate generator for the 8251 USART and its output is connected directly to the transmit and receive clock inputs of the 8251. The gate input to timer zero (GO) is tied active (high) to permanently enable this counter. The output of timer one (O1) is connected to the input of timer Ø2 (CS) and the two timers can be used together to form a 32 bit counter/timer. The gate inputs, count inputs, and count outputs are available for use at the SBC 880 connector (J2). Refer to Section VIII for a description of the 8253 functions.

8251 USART INTERFACE

Timer zero of the 8253 devides the 2 MHz clock down and provides the transmit and receive clocks (TXC and RXC) to the 8251. Transmit data ~~at~~ from the RS232 connector (J1 pin2) is level shifted by a 1489 RS232 receiver and applied to the receive data input (RXD) of the 8151. Transmit data (TXD) from the 8251 is level shifted by a 1488 RS232 (J1 Pin3). The reverse channel transmit (Pin 11 of J1) input at the RS 232 connector is level shifted by a 1489 RS232 receiver and provided to the DATA SET READY DSR/ input of the 8251. This input to the 8251 can be sensed as a status bit in the status register and has no other effect on the operation of the 8251. This allows programs run by the SBC 880 to sense not ready or buffer full conditions on serial I/O devices. The CTS/ input of the 8251 is tied active (low) permanently at the input of the 8251. This tells the 8251 that the RS232 interface is always ready to transmit data. The actual part used for the USART will be an 8251A or 9551. The older 8251 will not be used on the SBC 880. Refer to Section VII for a functicnal description of the USART device used on the SBC 880.

OUGHT TO BE PIN 6 OF J1?

NOT RJ232?

The RS 232 connector is configured to allow direct connection to a device without modems. Any modem signals required by the connected device will be satisfied by jumpers on the SBC 880. The SBC880 RS232 connector jumpers together the following RS 232 signals at the connector. Request to send is jumpered to Clear to send (Pins 4 and 5). Data terminal ready is jumpered to Data set ready and Carrier detect (Pins 6, 8 and 20). There is no software REQ/ACK therefore.

BAUD RATE DEVISORS

When the baud rate timer is initialized a divisor must be selected that will divide the 2 MHz clock to a frequency that is 16 times the baud rate desired. The following list of baud rate divisors will be of help in selecting the one for your application.

Baud Rate	Divisor
9600	13
4800	26
2400	52
1200	104
600	208
300	417
150	833
110	1136

```
                                PORTS
0003                  ICIL      =          03H      ;TIMER MODE WORD WRITE
0000                  I0        =          00H      ;BAUD RATE TIMER #0
0001                  I1        =          01H      ;TIMER 1
0002                  I2        =          02H      ;TIMER 2
0006                  CONDIA    =          06H      ;USART DATA REGISTER
0007                  CONCIL    =          07H      ;USART CONTROL REGTSTER

                      ;THE FOLLOWING CODE INITIALIZES THE USART AND TIMER.
                      ;THE BAUD RATE TIMER IS SET TO PRODUCE A TIMES 16
                      ;CLOCK FOR THE USART. THE BAUD RATE DESIRED IS 9600
                      ;SO THE DESIRED USART CLOCK RATE IS 16 TIMES 9600
                      ;OR 153,600 HZ.  A 2MHZ CLOCK IS DIVIDED DOWN BY
                      ;THE BAUD RATE TIMER TO DEVELOP THE USART CLOCK.
                      ;IN THE FOLLOWING EXAMPLE THE TIMER IS PROGRAMMED
                      ;TO RUN IN DECIMAL MODE AND GENERATE A SQUARE
                      ;WAVE.  A DIVISOR OF 13 IS USED WHICH YEILDS A USART
                      ;CLOCK OF 153,846 HZ.  THIS REPRESENTS A FREQUENCY
                      ;ERROR OF LESS THAN TWO TENTHS OF A PERCENT.

0000    3E36          INIL:     MVI        A,36H    ;WRITE TIMER MODE WORD
0002    D303                    OUT        ICIL                    (PORT 03)
0004    3E0D                    MVI        A,13     ;SET BAUD RATE (DIVISOR)
0006    D300                    OUT        I0
0008    3E00                    MVI        A,0
000A    D300                    OUT        I0
000C    3E7A                    MVI        A,7AH    ;WRITE USART MODE WORD
000E    D307                    OUT        CONCIL
0010    3E37                    MVI        A,37H    ;WRITE USART COMMAND WORD
0012    D307                    OUT        CONCIL

                      ;THE FOLLOWING IS AN EXAMPLE OF AN OUTPUT
                      ;ROUTINE FOR THE USART.
                      ;THE ROUTINE WILL TRANSMIT THE CHARACTER THAT
                      ;IS IN THE C REGISTER AND RETURN WITH THE
                      ;CHARACTER IN THE A REGISTER.

0014    DB07          CONOUT:   IN         CONCIL   ;TEST IF XMIT BUFFER EMPTY
0016    E601                    ANI        01H
0018    CA 0014                 JZ         CONOUT   ;LOOP UNTIL EMPTY
001B    79                      MOV        A,C      ;MOVE CHARACTER FROM C TO A
001C    D306                    OUT        CONDIA   ;TRANSMIT CHARACTER
001E    C9                      RET                 ;DONE

                      ;THE FOLLOWING IS AN EXAMPLE OF AN INPUT ROUTINE
                      ;FOR THE USART.
                      ;THE ROUTINE RECEIVES ONE CHARACTER FROM THE USART
                      ;AND RETURNS WITH THE CHARACTER IN THE A REGISTER.

001F    DB07          CONINP.   IN         CONCIL   ;TEST FOR CHARACTER RECEIVED
0021    E602                    ANI        02H
0023    CA 001F                 JZ         CONINP   ;LOOP UNTIL CHARACTER RECEIVED
0026    DB06                    IN         CONDIA   ;READ DATA CHARACTER FROM USART
0028    B7                      ORA        A        ;RESET THE CARRY FLAG
0029    C9                      RET                 ;DONE
```

```
                     ;THE FOLLOWING ROUTINE TESTS TO SEE IF A CHARACTER HAS
                     ;BEEN RECEIVED BY THE USART.
                     ;IF A CHARACTER IS PRESENT THE ROUTINE RETURNS WITH A
                     ;"FF" IN THE A REGISTER.
                     ;IF A CHARACTER HAS NOT BEEN RECEIVED THE ROUTINE WILL
                     ;RETURN WITH A "00" IN THE A REGISTER.

002A    DB07         STSCON: IN      CONCTL  ;TEST IF CHARACTER RECEIVED
002C    E602                 ANI     02H
002E    3EFF                 MVI     A,0FFH  ;PUT "FF" IN THE A REGISTER
0030    C0                   RNZ             ;RETURN IF CHARACTER RECEIVED
0031    2F                   CMA             ;CHANGE "FF" TO "00"
0032    C9                   RET             ;RETURN WITH "00" IN A

                     .END
```

00 = NULL
FF = undefined

PARALLEL I/O PORTS

The parallel output port is implemented with a 74LS374 edge-triggered register. The register outputs of the 74LS374 are buffered on the chip and need no additional buffering. The clock to the output port register is provided to the I/O connector (J2). Output data is latched at the rising (low to high) transition of this clock. This clock will transition every time the output port is selected during an I/O output cycle.

The parallel input port is implemented with a 74LS373 octal transparent latch. The tri-state buffers on this chip provide data directly tc the Z80 CPU bi-directional data bus when the input port is selected during an I/O input cycle. The data present at the latch inputs when the latch select strobe goes from high to low (input port selected) will be latched and then presented to the the Z80 data bus. The latch strobe signal is made available at the I/O connector.

ALL OPTIONS ON PCB TRACES ARE OPEN

SECTION VI BOARD OPTIONS

OPTION I - ON BOARD EPROM

A to H connection is $\overline{\text{EPS}}$

2708 EPROM

The SBC 2/4 comes etched for the 2708 EPROM. The options are as follows:

A to B connected = EPROM wait state generator enabled

1) $\overset{+5}{\text{Z}}$ to K Open
2) $\overset{\overline{\text{WR}}}{\text{Y}}$ to G Open
3) $\overset{-5}{\text{F}}$ to $\overset{\text{PIN21 VPP}}{\text{G}}$ Connected   PCB TOP
4) $\overset{\text{EPS}}{\text{H}}$ to $\overset{\overline{\text{OE}}}{\text{I}}$ Connected   PCB TOP
5) $\overset{+12}{\text{J}}$ to $\overset{\text{VDD}}{\text{K}}$ Connected   PCB TOP
6) L to M Open
7) $\overset{\text{COMP}}{\text{P}}$ to $\overset{\text{A10}}{\text{O}}$ Connected   PCB TOP
8) P to $\overset{\text{GND}}{\text{N}}$ Open
9) L to K Open
10) H to $\overset{\overline{\text{CE}}}{\text{M}}$ Open
11) V to W Connected   PCB BOTTOM
12) X to V Open

TMS2716 (TI 3 voltage EPROM). The options are as follows:

1) F to G Connected
2) H to I Connected
3) J to K Connected
4) L to M Connected
5) P to O Open
6) P to N Connected
7) Switch 6 of SW 3 Closed (A10)
8) L to K Open
9) H to M Open
10) V to W Connected
11) X to V Open
12) Y to G Open
13) Z to K Open

Intel type 2716 (+5 volt only EPROM). The options are as follows: 2K×8

PROM PINS

S 1) F to G Open — -5 ~~open~~

S 2) H to I Connected — (U) $\overline{EPS}$ to OE

S 3) J to K Open — +12 open

S 4) L to M Open — (K) ~~A10~~ IC ~~open~~

D 5) L to K Connected — A10 to A10

S 6) H to M Connected — (U) $\overline{EPS}$ to $\overline{CE}$ — 'EPROM SELECT'

U30 page 1 D 7) P to O Open

D 8) P to N Connected — GNDS B6 OF PROM SELECT COMP

PROMSEL D 9) Switch 6 of SW3 Closed (A10) — GNDS A10 ADDRESS COMP.

} THUS A10 ~~DOES~~ NOT GO INTO ADDRESS COMPARATOR NOTE OPTION 5) FOR A10

U10 D 10) V to W Connected } SMEMR TO U10 (MREQ TO U10) — 'READ ONLY' TO $\overline{EPS}$

D 11) X to V Open

PROM D 12) Y to G Open

D 13) Z to K Open — (+5 PULLUP TO PIN 22) Z TO G

4802 (~6116 NMOS) 2K×8

OPTION 2 - 4118 type RAM substituted for EPROM — 1K×8

The options are as follows:

2K×8 · 4802

ok 1) F to G Open — no -5

ok 2) H to I Connected — $\overline{EPS}$ to OE and CE

ok 3) J to K Open

ok 4) L to M Open

A10 to A10 CONNECTED 5) L to K Open

ok 6) H to M Connected — ← USE A INSTEAD OF H

COMPARATOR [illegible] OPEN TO A10 7) P to O Connected — A10 to PROM SELECT COMP ([illegible] OPEN)

COMPARATOR CONNECTED TO GND [illegible] 8) P to N Open

ok 9) V to W Open — 6-SW3 CLOSED TO GND

ok 10) X to V Connected — MREQ TO U10 : 'R & W' TO $\overline{EPS}$

ok 11) Y to G Connected — $\overline{WR}$ to pin 21

OPEN 12) Z to K Connected — +5 to (chip A10 ?) pin 19 1K×8 only

4802: 11 bits = 2K Therefore leave A10 (6-SW3) closed to gnd, and select address with A11-A15. Ensure that at least one bit A11-A15 of SW2 differs from SW3

OPTION 3 - Power on Jump, No Phantom Mode

The on board EPROM (Option 1) must be present to use the power on jump option. The Z80 board comes with options etched for 2708 EPROM and power on jump with no phantom mode. The options are set up as follows for power on jump with no phantom mode:

U17 $\overline{\text{RESET}}$ $\overline{\text{RESET}}$ FROM $\overline{\text{PRESET}}$

1) T to U Connected

2) Q' to R Connected

SMEMR AND ADDRESS MATCH TO U17 $\overline{\text{PRESET}}$ INPUT TO $\overline{\text{EPS}}$ GATE

OPTION 4 - Power on Jump with Phantom Mode

The on board EPROM (Option 1) must be installed to use this option. The options are set up as follows:

1) T to U Connected

2) Q to R Open

OPTION 5 - No Power on Jump

The EPROM or optional RAM (Option 1) may be used with this option installed. The option is set up as follows:

1) T to U Open

2) Q to R Connected

OPTION 6 - No EPROM

To disable address selection of the EPROM entirely, set up the options as follows:

1) Q to R Open

2) T to U Open

OPTION 7 - MWRT generated by Z80

The SBC 880 comes with this option etched on the board. The option is set up as follows:

GND BUFFER $\overline{EN}$

1) C to E Connected

ENABLES MWRT BUFFER

2) D to E Open

STATUS DISB

OPTION 8 - MWRT generated by external device

The option is set up as follows:

1) C to E Open

MWRT BUFFER ALWAYS DISABLED

2) D to E Open

OPTION 9 - MWRT generated by Z80 and external devices

1) C to E Open

2) D to E Connected DISABLES MWRT BUFFER BY $\overline{SDSB}=0$

SECTION VI

USART 8251 or 9551
OPERATION AND PROGRAMMING

The computer program controlling the USART performs these tasks:

1. Outputs control codes
2. Inputs status
3. Outputs data to be transmitted
4. Inputs data that has been received

Control codes determine the mode the USART wil] operate in and they are used to set or reset control signals output by the USART.

The status register contents will be read by the program monitoring the USART operation in order to determine error conditions, when and how to read data, write data or output control codes. Program logic may be based on reading status bit levels, or control signals may be used to request interrupts.

INITIALIZING

The USART may be initialized following a system reset or prior to starting a new serial I/O sequence. The USART must be reset following power up and subsequently may be reset at any time following completion of one activity and preceeding a new set of operations. Following a reset, the USART enters an idle state in which it can neither transmit nor receive data.

The USART is initialized with two, three or four control words from the processor. Figure 1 shows the sequence of control words needed to initialize the USART for synchronous or for asynchronous operation. For asynchronous operation a mode control is output to the device followed by a command. For synchronous operation, the mode control is followed by one or two SYNC characters, and then a command.

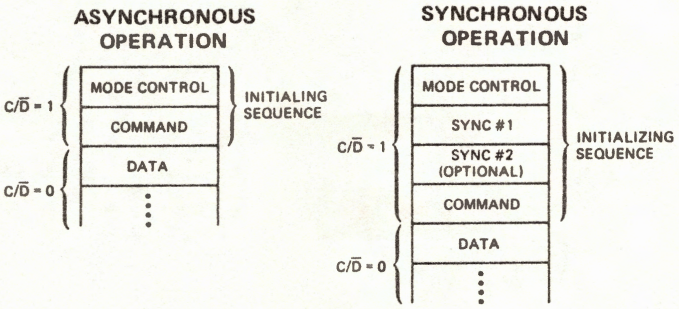


FIGURE 1. Control Word Sequence for Initialization.

Only a single address is set aside for mode control bytes, command bytes and SYNC character bytes. For this to be possible, logic internal to the chip directs control information to its proper destination based on the sequence in which it is received. Following a reset, the first control code output is interpreted as a mode control. If the mode control specifies synchronous operation, then the next one or two bytes (as determined by the mode byte) output as control codes will be interpreted as SYNC characters.

For either asychronous or synchronous operation, the next byte output as a control code is interpreted as a command. All subsequent byte output as control codes are interpreted as commands. There are two ways in which control logic may return to anticipating a mode control input; following an external reset signal or following an internal reset command.

MODE CONTROL CODES

The USART interprets mode codes as shown in Figure 2 and 3. Control code bits 0 and 1 determine whether synchronous or asynchronous operation is specified. A non-zero value in bits 0 and 1 specifies asynchronous operation and defines the relationship between data transfer, baud rate and receiver or transmitter clock rate. Asynchronous serial data may be received or transmitted on every check pulse, on every 16th clock pulse, or every 64th clock pulse. A zero in both bits 0 and 1 defines the mode of operation as synchronous.

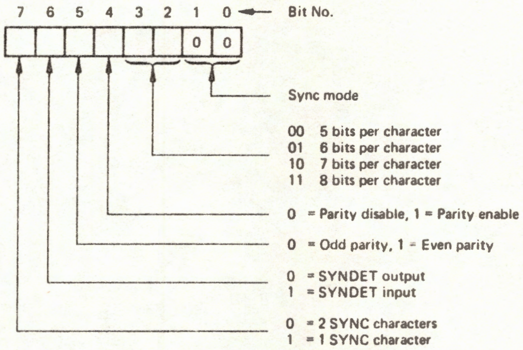


FIGURE 2. Synchronous Mode Control Code.

For synchronous and asynchronous modes, control bits 2 and 3 determine the number of data bit which will be present in each data character.

For synchronous and asynchronous modes, bit 4 and 5 determine whether there will be a parity bit in each character, and if so, whether odd or even parity will be adopted. Thus in synchronous mode a character will consist of five, six, seven or eight data bits, plus an optional parity bit. In asynchronous mode, the data unit will consist of five, six, seven or eight data bits, an optional parity bit, a preceeding start bit, plus 1, 1½, or 2 trailing stop bits. Interpretation of subsequent bits differs for synchronous or asynchronous modes.

Control code bits 6 and 7 in asynchronous mode determine how many stop bits will trail each data unit. 1½ stop bits can only be specified with a 16 x or 64 * baud rate factor. In these two cases, the half stop bit will be equivalent to 8 or 32 clock pulses, respectively.

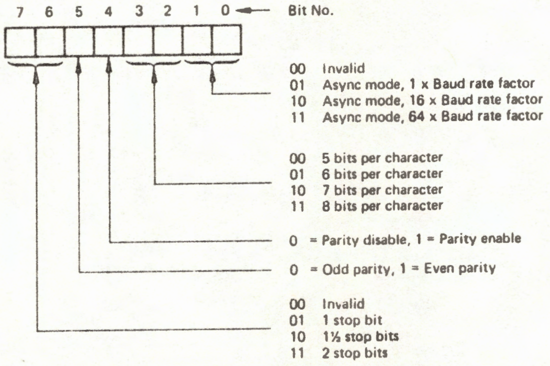


FIGURE 3, Asynchronous Mode Control Code.

In synchronous mode, control bits 6 and 7 determine how character synchronization will be achieved. When SYNDET is an output, internal synchronization is specified; one or two SYNC characters, as specified by control bit 7, must be detected at the head of a data stream in order to establish synchronization.

COMMAND WORDS

Command words are used to initiate specific functions within the USART such as, "reset all error flags" or "start searching for SYNC". Consequently, command words may be issued at anytime during the execution of a program in which specific functions are to be initiated within the communication circuit. Figure 4 shows the formats for the command words.

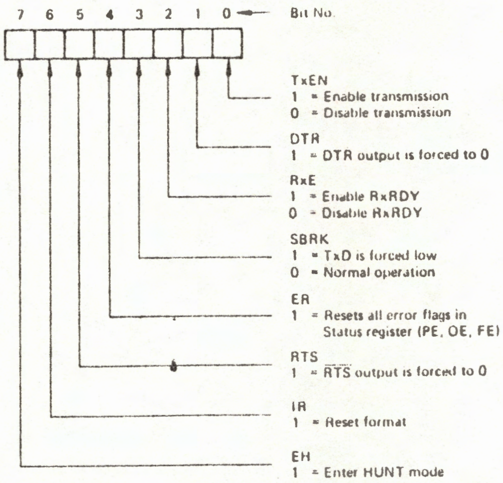


FIGURE 4. Control Command.

Bit 0 of the command word is the transmit enable bit (TxEN). Data transmission from the USART cannot take place unless TxEN is set in the command register. Figure 5 defines the way in which TxEN, T x E and T x RDY combine to control transmitter operation.

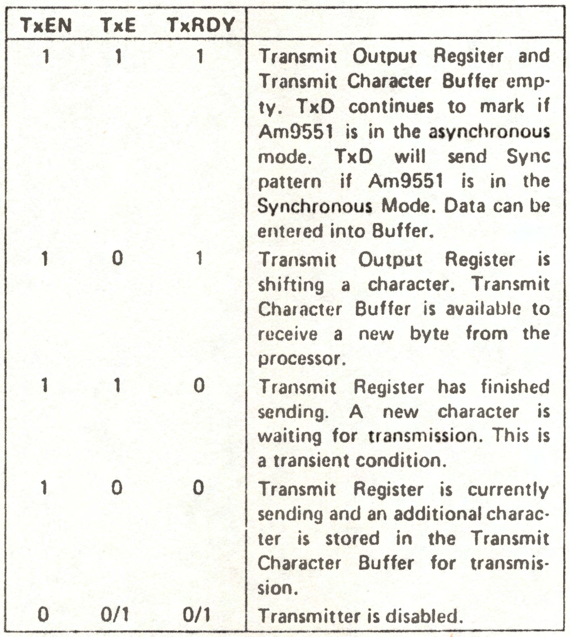
Bit 1 is the Data Terminal Ready (DTR) bit. When the DTR command bit is set, the DTR output connection is active (low). DTR is used to advise a modem that the data terminal is prepared to accept or transmit data.

Bit 2 is the Receiver Enable Command bit (R x E). R x E is used to enable the R x RDY output signal. RxE prevents the R x RDY signal from being generated to notify the processor that a complete character is framed in the Receive Character Buffer. It does not inhibit the assembly of data characters at the input, however. Consequently, if communication circuits are active, characters will be assembled by the receiver and transferred to the Receiver Character Buffer. If R x E is disabled, the overun error (OE) will probably be set; to insure proper operation, the overrum error is usually reset with the same command that enables R x E.

Bit 3 is the Send Break Command bit (SBRK). When SBRK is set, the transmitter output (T x D) is interrupted and a continuous binary "O" level, (spacing) is applied to the T x D output signal. The break will continue until a subsequent command word is sent to the USART to remove SBRK.

Bit 4 is the Error Reset bit (ER). When a Command Word is transmitted with the ER bit set, all three error flags in the Status Register are reset. Error Reset occurs when the Command Word is loaded into the USART. No latch is provided in the Command Register to save the ER command bit.

Bit 5 the Request To Send Command bit (RTS). Sets a latch to reflect the RTS signal level. The output of this latch is created independently of other signals in the USART. As a result, data transfers may be made by the Z80 to the Transmit Register; and data may be actively transmitted to the communication line through T x D regardless of the status of RTS.



TxEN	TxE	TxRDY	
1	1	1	Transmit Output Regsiter and Transmit Character Buffer empty. TxD continues to mark if Am9551 is in the asynchronous mode. TxD will send Sync pattern if Am9551 is in the Synchronous Mode. Data can be entered into Buffer.
1	0	1	Transmit Output Register is shifting a character. Transmit Character Buffer is available to receive a new byte from the processor.
1	1	0	Transmit Register has finished sending. A new character is waiting for transmission. This is a transient condition.
1	0	0	Transmit Register is currently sending and an additional character is stored in the Transmit Character Buffer for transmission.
0	0/1	0/1	Transmitter is disabled.

FIGURE 5. Operation of the Transmitter Section as a Function of TxE, TxRDY and TxEN.

Bit 6, the Internal Reset (IR) causes the USART to return to the idle mode. All functions within the USART cease and no new operation can be resumed until the circuit is reinitialized. If the operating mode is to be altered during the execution of the program, the USART must first be reset. Either the external reset connection can be activated or the Internal Reset Command can be sent to the USART. Internal Reset is a momentary function performed only when the command is issued.

Bit 7 is the Enter Hunt Command bit (EH). The Enter Hunt Mode Command is only effective for the USART when it is operating in the synchronous mode. EH causes the receiver to stop assembling characters at the R x D input and start searching for the prescribed sync pattern. Once the "Enter Hunt" mode has been initiated, the search for the sync pattern will continue indefinitely until EH is reset when a subsequent command word is sent, when the IR command is sent to the USART, or when SYNC characters are recognized.

STATUS REGISTER

The Status Register maintains information about the current operation status of the USART. Figure 6 shows the format of the Status Register.

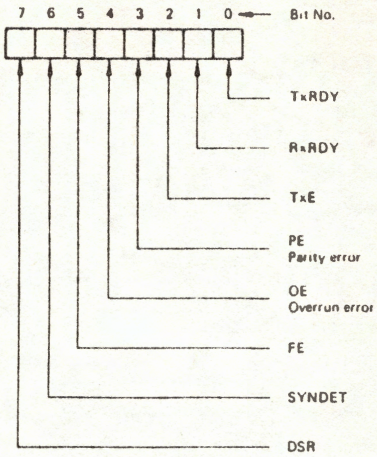


FIGURE 6. Status Register.

TxRDY signals the Z80 that the transmit character buffer is empty and that the USART can accept a new character for transmission.

RxRDY signals the Z80 that a completed character is holding in the Receive Character Buffer Register for transfer to the Z80.

TxE signals the processor that the Transmit Register is empty.

PE is the Parity Error signal indicating to the Z80 that the character stored in the Receive Character Buffer was received with an incorrect number of binary "1" bits.

OE is the receiver Overrun Error. OE is set whenever a byte stored in the Receiver Character Register is overwritten with a new byte before being transferred to the processor.

FE is the character framing error which indicates that the asynchronous mode byte stored in the Receiver Character Buffer was received with incorrect character bit format, as specified by the current mode.

SYNDET is the synchronous mode status bit associated with internal SYNC detection.

DSR is the status bit set by the external Data Set Ready signal to indicate that the communication Data Set is operational. All status bits are set by the function described for them. SYNDET is reset whenever the processor reads the Status Register. OE, FE, PE are reset only by command.

SECTION VIII

PROGRAMMABLE INTERVAL TIMER

INTRODUCTION

The Programmable Interval Timer used with this processor board can do functions normally done by software timing loops, such as event counting, time out delays, variable frequency generation, real time clock. With a minimal amount of software overhead, the interval timer can free the Z80 of the task of counting and do it faster.

The Programmable Interval Timer has three separate 16-bit counters (Refer to Figure 1) that can count at rates up to 2 MHz. Counter 0 is dedicated as the programmable baud rate generator for the USART. Counters 1 and 2 are available for use by the programmer.

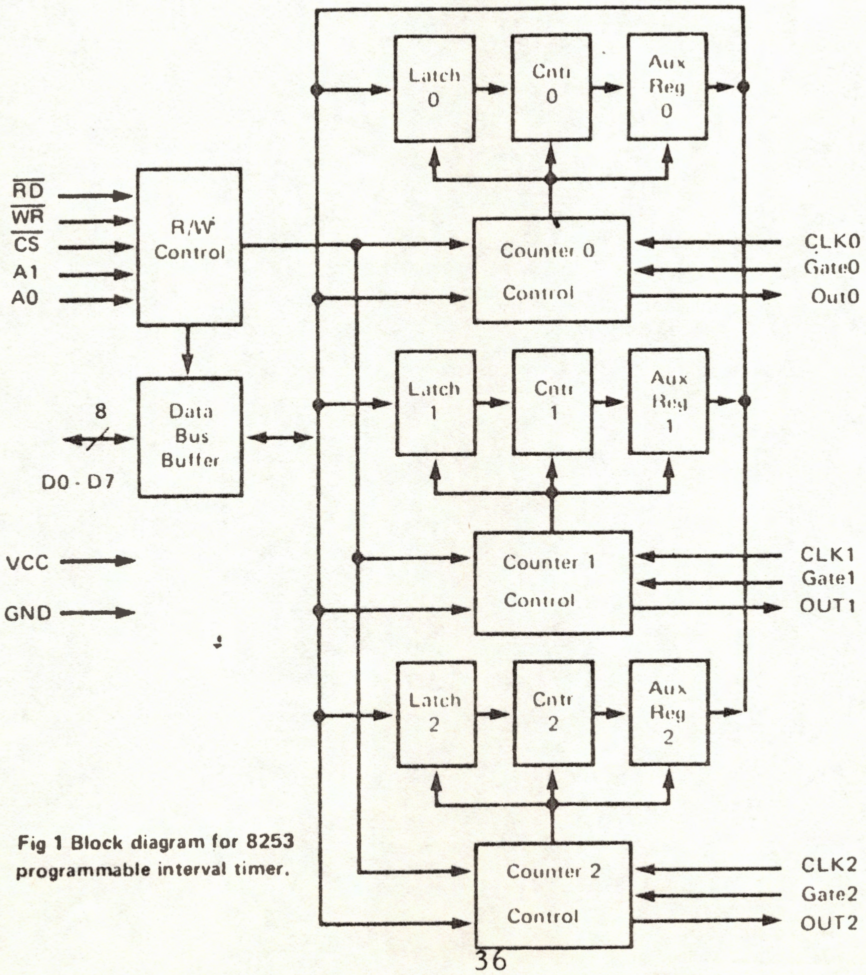


Fig 1 Block diagram for 8253 programmable interval timer.

The counters can be operated in six different modes:

Mode	0	Interrupt on Terminal Count
	1	Programmable One-Shot
	2	Rate Generation
	3	Square Wave
	4	Software Triggered Strobe
	5	Hardware Triggered Strobe

The counters count in binary or BCD in repetitive and single event modes - all synchronous to the processor clock.

PROGRAMMING

Associated with each counter is one 6-bit write-only control word register and two 8-bit write-only counter latches. To program the counter you initialize the control register and then program the counter latches. The counters can be programmed in any order, as long as each control word is programmed before the counter latches for that particular counter. (TABLE 1)

A1	A0	
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word

TABLE 1 Counter Addressing

CONTROL WORD REGISTER

The 6 bit control word register controls the counter mode and read/write sequencing of the counter latches. When A0 and A1 are both high D6 and D7 select the control register for each counter. (TABLE 2)

A1	A0	D7	D6	
1	1	0	0	Counter 0 CW
1	1	0	1	Counter 1 CW
1	1	1	0	Counter 2 CW
1	1	1	1	Illegal

TABLE 2 Control Word Register

It appears that only one register is being programmed because the address A1 and A0 is the same for all control registers. in actuality the upper two bits (D7 and D6) of the data word select the individual registers. The lower six bits (D5,D4,D3, D2,D1,D0) are register information. (TABLE 3)

D5	D4	D3	D2	D1	D0
RL1	RL0	M2	M1	M0	BCD =1 4 DECADES

RL1	RL0	
0	0	Counter latching command
0	1	Read/Load LSB latch
1	0	Read/Load MSB latch
1	1	Read/Load LSB, then MSB latch

TABLE 3 Control Register Format

		D3 M2	D2 M1	D1 M0	
MODE	0	0	0	0	
	1	0	0	1	
	2	X	1	0	(2 OR 6)
	3	X	1	1	(3 OR 7)
	4	1	0	0	
	5	1	0	1	

RLO and RL1 (D4 and D5) of the control word determine how the two counter bytes are to be accessed when the counter address is selected. They are also decoded to send a special instruction which latches the counter contents. MO, M1 and M2 (D1,D2,D3) of the control word determine in which of the six modes the counter is to operate. BCD (D0) selects binary or BCD counting. (TABLE 3)

COUNTER LATCHES

A0 and A1 in conjunction with RL0 and RL1 access the counter latches. A0 and A1 determine which of the three pairs of counter latches are to be accessed; the read/load (RL) bits of the control word register determine the upper/lower byte selection. If only RL0 is set, the least significant byte is being programmed. If only RL1 is set then the MSB is being programmed. If both bits are set, then a sequence of two write programs the LSB and then the MSB latch. Using this read/load format then requires the performance of two writes in sequence, if the device is to operate correctly.

It is not necessary to program all 16 bits of a counter, when either the lower or upper byte is a zero. Both latches are automatically cleared when the control word is programmed and remain zero until otherwise programmed.

OPERATING MODES

There are six available modes of counting. Modes 2 and 3 are repetitive and all others are single event modes. TABLE 4 contains a gate summary for the different modes.Figure ? 2 shows the timing for the six modes. See Intel DataBook

MODE 0 INTERRUPT ON TERMINAL COUNT

In this mode a control word write or writing to any counter latch forces the output low. After the write to the counter is completed, it begins counting. At the completion of the count (Counter Equals zero), the output goes high and remains high until a new control word or count is loaded. Reloading the counter latches during counting suspends the current count. At the end of reloading, the counter begins counting with the new divisor, the gate input suspends counting when low, and enables counting when high.

MODE 1 PROGRAMMABLE ONE-SHOT

In this mode, the output is high when the counter is not counting. Arising transition of the gate input triggers the counter to begin counting which forces the output low. Upon completion, the output goes high. Since the counter is retriggerable, any rising edge on the gate causes the counter to restart at the beginning. The counter can be reloaded at anytime. Any subsequent trigger initiates the new count.

MODE 3 SQUARE WAVE

Similar to Mode 2, except that the output remains high for half the count and low for half the count for even divisors, for odd numbers, the output is high for (N+1)/2 counts and low for (N+1)/2 counts. In other words, the remainder of division by 2 is added to the output high time. If the counter is reloaded while counting, the new divisor becomes effective after the next output transaction. The gate input functions identically to Mode 2.

MODE 4, SOFTWARE-TRIGGERED STROBE

In this mode, the output is normally high. Loading the counter latch (es) initiates counting. If counting is in progress at the time of the load, the current count runs to completion and the subsequent count reflects the new value. Upon completion, the output goes low for one clock period.

MODE 5 HARDWARE TRIGGERED STROBE

This mode is the same as MOde 1, except that the output is normally high and goes for one clock period upon completion of counting.

COUNTER READING AND LATCHING

The counters can be read in two ways. In one of them, issuing a normal read to the counter's specified address transfers the counter outputs directly to the data bus. If the counter is counting, the contents are changing continuously. For an assured reading of the actual counter contents, the counter must be inhibited by disabling the clock or alternatively by forcing the gate low, if it is in modes 0 or 4. Note that the counter latches are write-only and that the counter itself is read. In reading, as with writing, the read/load bits of the control word register determines the accessing of the counter contents.

The second method of reading the counter uses the counter latching command. Issued like a controlword, this command performs a counter latching operation: Freezing the contents in an auxiliary register and giving a stable, readable value. Once latched, the contents can be read out at any time without affecting the counter operation. In operation the user simply issues the latch command for the particular counter (Tables 5) at the desired point in time to latch the current contents. The saved counter contents are now read as though one were reading the counter latches.

NOTE: The latch command does not affect the programmed read/load format or mode, so that the bytes read remain as previously programmed by the control word.

MODE	0	1	2	3	4	5
↑ initiate count		x	x	x		x
low inhibit count	x		x	x	x	

TABLE 4 GATE SUMMARY

A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	
1	1	0	0	0	0	x	x	x	x	Latch counter 0
1	1	0	1	0	0	x	x	x	x	Latch counter 1
1	1	1	0	0	0	x	x	x	x	Latch counter 2

TABLE 5 LATCH COMMAND

APPENDIX

TABLE 1

Connector J1 Signals

Pin No.	Function
2	RS232 Transmit Data
3	RS232 Receive Data
4	Request to Send
5	Clear to Send
6	Data Set Ready
7	Signal Ground
8	Carrier Detect
11	Reverse Channel Transmit
20	Data Terminal Ready

TABLE 2

Connector J2 Signals

Pin No.	Function
1	Outport Port Data Bit 0
2	Outport Port Data Bit 1
3	Outport Port Data Bit 2
4	Outport Port Data Bit 3
5	Outport Port Data Bit 4
6	Outport Port Data Bit 5
7	Outport Port Data Bit 6
8	Outport Port Data Bit 7
9	Signal Ground
10	Output Port Clock
11	Counter 1 Gate Input
12	Counter 2 Gate Input
14	Input Port Data Bit 0
15	Input Port Data Bit 1
16	Input Port Data Bit 2
17	Input Port Data Bit 3
18	Input Port Data Bit 4
19	Input Port Data Bit 5
20	Input Port Data Bit 6
21	Input Port Data Bit 7

TABLE 2 (continued)

22	Signal Ground
23	Input Port Strobe
24	Counter 1 Output (also Counter 2 Input)
25	Counter 2 Output

SBC880 PROCESSOR BOARD PARTS LIST

Qty.	Location	Description			5V max MA	LS
1	U1	✓ 7420	Dual 4 Input Nand	✓	~~22~~ 11	2.2
2	U2, 21	7432	Quad 2 Input or		2x38	15.2
2	U3, 11	✓ 7474	Dual D Flip flop	✓	2x15	6
1	U4	✓ 7404		✓	33	6.6
1	U5	✓ 7402	Quad 2 Input NOR	✓	27	6.4
1	U6	1489	RS232 Receiver		26	—
1	U7	1488	RS232 Driver		2+@±12v	—
1	U8	8251A	Usart		80	—
1	U9	8253	Programmable timer		85	—
1	U10	✓ 7400	Quad 2 Input NAND	✓	22	4.4
2	U12, 13	74LS240	Octal Inverting Buffer		2x50	—
1	U14	✓ 7410	Triple 3 Input NAND	✓	16.5	3.3
1	U15	74LS373	Octal Transparent Latch		40	—
1	U16	74LS374	Octal Register		45	—
1	U17	74112	Dual JK Flip Flop	(LS)	8	—
2	U18, 20	7408	Quad 2 Input AND		2x33	13.2
1	U19	SBC-2/4	CPU Z80A		200?	—
4	U22, 27, 28, 33	74LS241	Octal Non-Inverting Buffer		4x54	—
1	U23	2708 OR 2716 OR 2516	EPROM) EPROM) Not included with kit EPROM)		100	
2	U24, 25	2114-3	1K ~~X4~~ static ram	300ns	2x70	
3	U26, 29, 30	8131	6 Bit Comparator		3x74	
2	U31, 32	✓ 74367	Hex Buffer = 8097	✓	2x77	—
1	VR1	✓ 7805	+5 volt regulator		10	
1	VR2	7812	+12 volt regulator	~~4884~~		
1	VR3	~~7905~~	~~-5 volt regulator~~	2708 only		
1	VR4	7912	-12 volt regulator	use L for 1488 only		
1	Y1	4MHz crystal				
1	2M, 4M	switch SPDT				
3	SW1,2,3	6 POS DIP Switch				
5	RP1, 3, 4,5,6	1K Resistor SIP Pack 8 pin			5x3½x5mA = 87½	
1	RP2	1K Resistor DIP Pack 14 pin			6½x5mA	32½
6	R1,3,4,6,7/8	1K Resistor ¼w carbon				
1	R5	330 Ohm Resistor ¼w carbon				
1	R2	4.7K Resistor ¼w carbon				
6	C1,2,3,4,9,11	1.5uf Capacitor Tantalum 25v				
16	C10,12,13,14,15, 16,17,18,19,20,21, 22,23,24,25,26	.1 uf Capacitor Ceramic Disk				
3	C5,6,7	.001 uf Capacitor Ceramic Disk				

1827.5mA

max w/LS = 1602.3

#1096.5

avg@60% (LS) 961.38

4V x 1A = 4w

BOARD ASSEMBLY INSTRUCTIONS

The SBC880 single board computer is intended for those people who have had some prior experience with kit building and digital electronics. If you do not fall into this category it is highly recommended that you find an experienced person to help you with the assembly and checkout of the board.

Although there is nothing sacred in the suggestsd steps that follow, if you will follow them step-by-step, you should find your task much easier. We suggest that you start at a time when you will be able to complete the board. It will help to mark the boxes as you complete each step.

□ 1. Make sure you have the tools you will need to assemble this kit. For this board you will need the following: a soldering iron (20 watts maximum). Rosin core solder (preferably 63/37), diagonal cutters, a small magnifying glass, a screwdriver, and a lead former or a pair of needle-nose pliers.

□ 2. Check the parts received against the parts list. Take special care to correctly identify look-alike parts, i.e., resistors, capacitors, regulators, sockets and IC's.

Do this now before you proceed any further.

CAUTION

Use eye protection while soldering or cutting wire.

□ 4. Install 20 pin sockets at U12, U13, U15, U16, U22, U27, U28, U33.
DO NOT solder in place.

□ 5. Install 18 pin sockets at U24, and U25.
DO NOT solder in place.

□ 6. Install 16 pin sockets at U17, U26, U29, U30, U31, U32.
DO NOT solder in place.

□ 7. Install 14 pin sockets at U1, U2, U3, U4, U5, U6, U7, U10, U11, U14, U18, U20 and U21.
DO NOT solder in place.

□ 8. Install 24 pin sockets at U9 and U23.
DO NOT solder in place.

□ 9. Install 28 pin sockets at U8.
DO NOT solder in place.

□ 10. Install 40 pin socket at U19.
DO NOT solder in place.

1	C8	100 uf Capacitor Electrolytic
1	J1	26 pin Header RS232
1	J2	26 pin Header I/O Port
4	VR1,2,3,4	6/32 Machine Screw
4	VR1,2,3,4	6/32 Hex Nut
4	VR1,2,3,4	#6 Washer
1	VR1	Heat Sink
13	U1,2,3,4,5,6, 7,10,11,14,18, 20,21	14 pin L.P. Sockets
6	U17,26,29,30 31,32	16 pin L.P. Sockets
2	U24,25	18 pin L.P. Sockets
8	U12,13,15, 16, 22,27,28,33	20 pin L.P. Sockets
2	U9,23	24 pin L.P. Sockets
1	U8	28 pin L.P. Sockets
1	U19	40 pin L.P. Sockets
1	P.C.	Printed Circuit Board

☐ 11. A handy trick to help you construct your board is to insert all the above sockets into the board first, then place the flat styrofoam cover you received with your kit box firmly against the top of the board. Turn it over, holding the flat styrofoam piece tightly against the board. The IC sockets should notw be on the bottom. Press the board down, forcing the sockets into the styrofoam. Now solder alternating corner pins of the IC sockets to hold them in place. Now turn the board over and very carefully inspect it to determine that all IC sockets are down flat against the board. If you find any that are not down flat, melt the solder joints at the corners of the IC socket and press it down against the board.

When you have determined that all IC sockets are down firmly on the board, turn the board back over and solder all the pins. Make sure that all the pins are sticking through the board. IC sockets are very difficult to remove once they are soldered onto a board.

22. 1. 1. 1.

☑ 12. Install 1.5 ufD capacitors at C1, C2, ~~C3~~, C4, C9 and ~~C11~~. Be sure the + and – polarity is observed.

☑ 13. Install .1 ufD capacitors at C10, C12 through C26.

☑ 14. Install the three DIP switches at SW1, SW2, and SW3. DO NOT place these DIP switches in sockets – they are liable to fall out under use.

☑ 15. Install the 100 microfarad electrolytic capacitor at C8.

☑ 16. Install the .001 ufD capacitor at C5, C6, and C7.

☑ 17. Install the 330 ohm resistor (orange, orange, brown) at R5.

☑ 18. Install 1K ohm resistors (black, brown, red) at R1, R3, R4, R6, R7 and R8.

☑ 19. Install the 4.7K ohm resistor at R2.

☑ 20. Install 1K 8 pin SIP resistor packs at RP1, RP3, RP4, RP5, and RP6. RP6 is located just above the edge connector.

☑ 21. Install 1K 14 pin DIP resistor pack at RP2.

☑ 22. Install a 4MHz crystal at Y1.

☑ 23. Install the voltage regulators. The 7805/LM340T5 regulator is installed at VR1. This is the plus 5 volt regulator and should be used with a heat sink. If you have a good heat sink compound, we suggest you use it at this regulator only. Use it sparingly, as too much is worse than none at all. Now install the 7812/LM340T12 (plus 12 volts) regulator at VR2. Install the 7805/LM320T5 (minus 5 volts) regulator at VR3. Install 7912/LM320T12 (minus 12 volts) regulator at VR4. Button them down with the screws and nuts supplied. Place the screw through the solder side of the board with the nut next to the regulator.

☐ 24. Install header pins at J1 and J2.

□ 25. Install a single-pole double-throw switch at 2M/4M position. If a switch has not been included in your kit solder in the wire-wrap pins in its place the center to top hole selects a 2 MHz clock. The center to bottom hole selects a 4MHz clock. Regardless of the switch setting the clock number signal on pin 49 of the S100 bus will be 2MHz as per IEEE specifications.

□ 26. Check all your solder joints carefully. Inspect the board for cold solder joints for solder bridges.

□ 27. BEFORE INSTALLING ANY IC's – place the board, in your computer and check all the voltages to make sure that you do not have any power supply shorts on the board. The output voltages from all the regulators can be measured on the pin facing toward the top of the board (away from the S100 connector). Be careful not to let your probes short the voltage regulator pins together. Since this can destroy a voltage regulator very neatly – and quickly. If all of the voltages are up to par (plus or minus about half a volt or so), continue to step 28, otherwise, check the board again for shorts. Find the short before you install any IC's and connect it.

□ 28. Install all IC's in locations shown on the assembly drawing.

□ 29. Install whatever options you have chosen from the options list.

□ 30. You should now be on the air.